UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/816,796	03/23/2001	Paul E. McKenney	BEA9-2001-0001-US1	5819

49056 7590 09/20/2006

LIEBERMAN & BRANDSDORFER, LLC
802 STILL CREEK LANE
GAITHERSBURG, MD 20878

EXAMINER	
MOLL, JESSE R	

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 09/20/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/816,796	MCKENNEY, PAUL E.	
	Examiner	Art Unit	
	Jesse R. Moll	2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *05 July 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
9/18/2006

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claims 12-25 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention. Claims 12 and 22 recite the limitation "A processor for use in a computer readable medium of a multiprocessor computer system". The specification does not describe how a processor can be used in a computer readable medium. A computer readable medium typically is only used to store and load data. The specification makes no mention of how to use a processor in such a device or even how to put a processor in the device.

1. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

2. Claims 1-25 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 1 recites the limitation "said shared resource element" in line 5. There is insufficient antecedent basis for this limitation in the claim. Examiner assumes for the purpose of examination that the limitation read "said shared resource". The claim also recites the limitation "storing said pointer" in line 9. Examiner requests the limitation read "said storing said pointer". It is unclear whether the storing references the previously said storing.

Further regarding claim 1, it is unclear how a method (which is merely an action of doing something) comprises instructions (which are objects).

Claims 2-11 are rejected due to their dependence on indefinite parent Claim 1.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claim 1 attempts to claim both an apparatus (instructions) and process. Claims should be directed to a "process, machine, manufacture, **or** composition of matter".

Examiner recommends removing all instances of the term "instructions for". Claims 2, 3 and 7-11 have similar issues with the limitation "instructions for".

> Claims to computer-related inventions that are clearly nonstatutory fall into the same general categories as nonstatutory claims in other arts, namely natural phenomena such as magnetism, and abstract ideas or laws of nature which constitute "descriptive material. "Abstract ideas, Warmerdam, 33 F.3d at 1360, 31 USPQ2d at 1759, or the mere manipulation of abstract ideas, Schrader, 22 F.3d at 292-93, 30 USPQ2d at 1457-58, are not patentable. Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of data structures and computer programs which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data (See MPEP section 2106, IV, B, i).

Claim 1 comprises steps of allowing, indicating, and forcing. They are just an abstract idea. The claim does not provide practical application that produces a useful, tangible and concrete result. Therefore, this claim is non-statutory. Similar problems exist in claims 2-11.

Claims 12 and 22 mainly comprises instructions which are software per se. The instructions are not described as stored in a computer readable medium to function with a computer to effect a practical application that results in a useful, tangible and concrete result. Therefore, the claims are non-statutory. Similar problems exist in claims 17-19, 21, 24 and 25. As written, the instructions merely exist in said medium but are not claimed in such a way to create a useful, tangible and concrete result. Nothing is claimed about the actions of the instructions, only their intended use.

Regarding claim 1, Examiner recommends replacing the limitation "forcing said write operations to non-local memory to precede storing... to said new element of said shared resource" in lines 8-10 to "executing said write operations to non-local memory prior to said storing... to said new element of said shared resource in response to said indicating" to add a tangible result to the claims.

Regarding claims 12 and 22, Examiner recommends replacing the limitation "A processor for use in a computer readable medium of a multiprocessor computer system, comprising:" with "A processor for use in a multiprocessor computer system executing instructions on a computer readable medium, said instructions comprising:" and removing all instances of the term "in said medium".

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-25 are rejected under 35 U.S.C. 102(b) as being anticipated by Robertson (U.S. Patent No. 5,850,632), hereafter referred to as Robertson'632.

Referring to claims 1 and 12, Robertson'632 discloses as claimed a method for maximizing CPU performance in a multiprocessor (see Fig. 2), comprising computer implemented instructions(a processor must contain instructions), comprising:

allowing write operations in local memory (each of the instruction caches 21, 26, 31, and 36, see Fig. 2, is broadly interpreted as a local memory) to execute in an arbitrary order (see col. 16, lines 31-34, regarding the caches being fully associative. Note each of the instruction caches 21, 26, 31, and 36, see Fig. 2, is individually and locally used by the associated processors 71-74, see Fig. 2; and a full associative replacement algorithm is used to have an arbitrary order to replace the cache lines thereof) and at any time prior to storing a pointer (PC, program counter) from an existing element of a shared resource (old instruction) to a new element of a shared resource (next instruction; stored by Program counter register in the Robertson'632's system) (note PC will be incremented or changed to a new number after the write operation; and when the program counter register saves a new PC value, it is interpreted as a new element); explicitly indicating a set of write operations to non-local memory (memory configuration cache 305, see Fig. 5, is broadly interpreted as a non-local memory since it is not disposed inside the processors 71-74, see Fig. 2) be conducted in a specified order (see col. 31, lines 38-57, regarding the specified order for writing (replacing) the cache lines of the memory configuration cache 305); and forcing said write operation to non-local memory (memory configuration cache 305, see Fig. 2, is broadly interpreted as a non-local memory since it is not disposed inside the processors 71-74, see Fig. 2) to precede storing said pointer from an existing element of a shared resource to said

new element of said shared resource (note the forcing step occurs when the Robertson'632's system also has a memory access (write operations) to memory configuration cache 305).

As to claims 2, 13, and 23, Robertson'632 also discloses: assigning first and second registers of a CPU for storing associated first and second instruction addresses (note the Robertson'632's processor certainly comprises registers such as PC (program counter), MAR (memory address register), General Data/Address Registers, or CAR (control address register for storing instruction addresses).

As to claim 3, Robertson'632 also discloses: providing a third instruction referencing said registers (this is the situation when the registers, such as PC (program counter), MAR (memory address register), General Data/Address Registers, or CAR (control address register) storing the first and second instruction addresses is referred to as the source or destination registers in a third instruction).

As to claims 4 and 14, Robertson'632 also discloses: said third instruction specifies ordering between said first and second instructions (this is the situation when the registers storing the first and second instruction addresses are referred to as the destination registers in a third LOAD instruction, therefore, certainly operating the ordering between said first and second instructions).

As to claims 5 and 15, Robertson'632 also discloses: said third instruction indicates said first instruction's execution attaining a first specified state of execution prior to said second instruction's execution attaining a second specified state of execution (note this occurs in the Robertson'632's system when either one of the first

instruction and the second instruction depends from the other and each instruction's execution invokes its state of execution specified by such as its opcode).

As to claims 6 and 16, Robertson'632 also discloses: said first and said second specified states of execution are selected from the group consisting of: committing instruction execution, initiating memory access, completing a memory access, initiating an I/0 access, completing an I/O access, and completing instruction execution (note this occurs in the Robertson'632's system when the first instruction and the second instruction is a memory load/store operation and it therefore certainly involves at least one of initiating memory access, completing a memory access as claimed).

As to claims 7, 17, and 24, Robertson'632 also discloses: assigning a sequence number to an associated instruction for maintaining instruction ordering (note this is the situation in the Robertson'632's system when a sequence of program to be executed is saved in the Robertson'632's main memory and each instruction in the program is assigned by a logical address or physical address number)

As to claims 8 and 18, Robertson'632 also discloses: statically encoding said sequence number within said instruction (inherently existing in the Robertson'632's processor when a sequence of program is therein).

As to claims 9 and 19, Robertson'632 also discloses: dynamically encoding said sequence number within said instruction (as set forth above, inherently the Robertson'632's processor comprises registers such as MAR (memory address register), General Data/Address Registers, or CAR (control address register for storing instruction addresses and for dynamically encoding the sequence number).

As to claims 10 and 20, Robertson'632 also discloses: placing a range of instructions into a hierarchical ordering system (note the control unit of the Robertson'632's CPU is reasonably and broadly interpreted as a hierarchical ordering system and a range of instructions is inherently placed in a process table).

As to claims 11, 21 and 25, Robertson'632 also discloses: implementing a special instruction for maintaining a hierarchical execution of said instruction (such as a microinstruction, existing in the Robertson'632's processor for control the instruction execution, which is broadly interpreted as a special instruction for maintaining a hierarchical execution).

Referring to claim 22, Robertson'632 discloses as claimed a processor for use in a multiprocessor computer system (see Fig. 2), comprising: a first instruction for allowing write operations in local memory (each of the instruction caches 21, 26, 31, and 36, see Fig. 2, is broadly interpreted as a local memory) to occur in an arbitrary order (see col. 16, lines 31-34, regarding the caches being fully associative. Note each of the instruction caches 21, 26, 31, and 36, see Fig. 2, is individually and locally used by the associated processors 71-74, see Fig. 2; and a full associative replacement algorithm is used to have an arbitrary order to replace the cache lines thereof), a second instruction for explicitly indicating a set of write operations to non-local memory (memory configuration cache 305, see Fig. 2, is broadly interpreted as a non-local memory since it is not inside the processors 71-74, see Fig. 2) to be conducted in a specified order (see col. 31, lines 38-57, regarding the specified order for writing (replacing) the cache lines of the memory configuration cache 305), wherein write

operations to non-local memory must execute prior to storage of a pointer (PC, program counter) from an existing element of a shared resource (old instruction) to a new element of a shared resource (next instruction; stored by Program counter register in the Robertson'632's system) (note PC will be incremented or changed to a new number after the write operation); and a third instruction for managing order of execution of said first and second instructions (note the above limitations are disclosed by Robertson'632 as set forth above in claim 4); wherein execution of said second instruction is responsive to said first instruction reaching a specified state of execution and said specified state of execution is selected from the group consisting of: committing instruction execution, initiating memory access, completing a memory access, initiating an I/O access, completing an 1/0 access, and completing instruction execution (note the above limitations are disclosed by Robertson'632 as set forth above in claims 5 and 6).

Response to Arguments

6. Applicant's arguments filed 05 July 2006 have been fully considered but they are not persuasive.

7. Applicant states:

> The Examiner has asserted that program counter register of Robertson is equivalent to the pointer claimed by Applicant. Applicant's amended independent claims clearly state storing a pointer from an existing shared resource to a new element of the shared resource. It is inherent that a shared resource is stored in non-local memory, since by its very nature a shared item has to be available in a non-local manner so that the objects sharing the item can access the item. To accomplish this, the Examiner asserts that the memory configuration cache of Robertson '632 is non-local memory. Based on this assertion by the Examiner, Robertson '632 must store the value

of the program counter in non-local memory, i.e. memory configuration cache. However, the memory configuration cache of Robertson stores memory configuration information for the memory configuration service unit. The memory configuration service unit of Robertson is an ASIC (application specific integrated circuit) which stores address ranges. See Col. 9, lines 18-21. There is no program counter in Robertson's memory configuration cache, i.e. non-local memory adapted to contain a shared resource, because Robertson's memory configuration cache stores memory configuration information for the memory configuration service unit, which does not have a program counter. Robertson's memory configuration service unit stores address ranges - not a program counter. Furthermore, the Examiner states that "the program counter is stored in the program counter register in the Robertson '632's system", See Office Action dated April 5, 2006 pages 12 and 13, and implies that the program counter register is in the system and not in non-local memory. Applicant's claims clearly state that the pointer pertains to a shared resource, i.e. non-local memory, which it inherently must be since it is a part of a shared resource. Clearly, Robertson does not expressly or inherently describe storing a pointer, or an equivalent thereof, to a new element of a shared resource in non-local memory. As indicated by the Examiner, the program counter of Robertson is stored in a program counter register in the system, not inherently in non-local memory. Applicant's claimed invention requires the use of the shared resource for storage of the pointer in non-local memory while Robertson does not utilize a shared resource for storage of a pointer or even a value of the program counter in non- local memory. Accordingly, Applicant' s invention is not anticipated by Robertson '632.

Examiner disagrees. Claim 1 merely requires that the pointer points to and from elements in a shared resource (program memory). A program counter instructs the processor which instruction to execute next. I therefore is a pointer which points from the current instruction (which is shared since a program can be run on two different processors) to the next instruction (which is also shared). The claim does not make any mention where the pointer itself is stored. The claim states “storing a pointer **from** an existing element of a shared resource **to** a new element of said shared resource”. The limitation can be reasonably interpreted as: storing a pointer anywhere which points from an existing element of a shared resource and points to a new element of said shared resource.

8. Applicant states:

With respect to the force element in Applicant's claims, Applicant requires the force to be associated with storing a pointer to an element of the shared resource. As discussed above, a

> shared resource is inherently stored in non-local memory, as such storage in non-local memory is required for the resource to be shared. The pointer of Applicant is associated with the shared resource, i.e. non-local memory. Storing a pointer, or a program counter based upon the Examiner's position, would again require Robertson to store the program counter in the memory configuration cache as the shared resource. However, as noted above, Robertson cannot store a program counter in the memory configuration cache. Accordingly, Robertson does not force a write operation in the manner as claimed by Applicant.

Examiner disagrees. As stated above, claim 1 does not express any limitation on where the pointer is stored, but merely limits the properties of the pointer which is stored.

9. Applicant states:

> Furthermore, Applicant's claims support that a write operation to non-local memory is forced to occur prior to an action of storing the pointer to a new element of the shared resource. To create an equivalent action in Robertson, the Examiner must not only define the memory configuration cache of Robertson as non-local memory, but also must define the steps Robertson will take based on an event that occurred as a "forced" operation. However, there is no support in Robertson for the "force" of the write operations to non-local memory to precede storing of the pointer to the new element of the shared resource, even if the non-local memory can be interpreted as memory configuration cache. To reach the Examiner's conclusion, the Examiner asserts that the term "force" is a broad term and is "interpreted as the action Robertson '632's system will take based on the event occurred." However, the use of the term force implies that this is an action that must be undertaken. There is no express or inherent force operation associated with the act of storing a pointer to a new element of a shared resource in Robertson.

Examiner disagrees. The term force merely states that the event happens. Processors operate on certain parameters. Computers are a finite state machine and as such always operate a certain way based on input and internal state. If an event occurs in a processor, the event must have occurred because of a set of rules the processor operates on. It can therefore be said that the event was forced by these rules. The claims make no mention of the relationship of the operations. While running a program on a processor, the program counter is updated whenever a new instruction is executed. Therefore the action of storing a pointer to a new element occurs after

every instruction. Since there is no relationship claimed between said storing and said write operations to non-local memory, the storing is interpreted as the update to the program counter which occurs after a write to the memory configuration cache 305.

10. Applicant states:

> "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.' " In re Robertson, 169 F.3d 743, 49 USPQ 2d 1949 (Fed. Cir. 1999), citing Continental Can Co. v. Monsanto Co., 948 F.2d 1264, 1268, 20 USPQ 2d 1746, 1749 (Fed. Cir. 1991). "The mere fact that a certain thing may result from a given set of circumstances is not sufficient." Id. The Examiner claims that Robertson uses "force" and refers Applicant to Figs. 2 and 5 of Robertson. However, the Examiner does not show how these figures show the use of force. In searching the entirety of Robertson, Applicant found only a single reference to the term "force", and this was limited to a "row access". See Col. 30, line 33. There is no reference in Robertson for the use of force in the manner claimed by Applicant.

Examiner disagrees. As stated above, the term force, with respect to computers, merely limits the action to occur. Since this event necessarily will occurs with normal operation of the system of Robertson (as shown above), the rejection under 3.5. U.S.C. 102 stands.

Conclusion

11. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jesse R. Moll whose telephone number is (571)272-2703. The examiner can normally be reached on M-F 9:00 am - 5:30 pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz M. Fleming can be reached on 571-272-4145. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
9/18/2006

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Jesse R Moll
Examiner
Art Unit 2181

JM 9/13/06

FRITZ FLEMING
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100
9/18/2006